**PLANO DE GERÊNCIA DE CONFIGURAÇÃO**



**QMS - Query Management System**

Autores: Carlos Henrique.

Última atualização: 04/10/17.

Versão: 3.0

**HISTÓRICO DE ALTERAÇÃO DO DOCUMENTO**

|  |  |  |  |
| --- | --- | --- | --- |
| **DATA** | **VERSÃO** | **DESCRIÇÃO** | **AUTOR** |
| 19/08/17 | 1.0 | Inicialização | Carlos Henrique |
| 27/08/17 | 2.0 | Adição de ferramentas | Carlos Henrique |
| 04/10/17 | 3.0 | Correção de termos | Carlos Henrique |

**SUMÁRIO**

[Introdução e Finalidade 3](#_Toc495023312)

[Referências 3](#_Toc495023313)

[Gerenciamento de Configuração de Software 3](#_Toc495023314)

[Organização, Responsabilidades e Interfaces 3](#_Toc495023315)

[Artefatos da Gerência de Configuração 3](#_Toc495023316)

[Ferramentas, Ambiente e Infra-estrutura 4](#_Toc495023317)

[Ferramentas 4](#_Toc495023318)

[Requisitos mínimos de Hardware do Ambiente de Desenvolvimento 5](#_Toc495023319)

[Requisitos mínimos de Hardware do Ambiente de Execução (smartphone) 5](#_Toc495023320)

[O Programa de Gerenciamento de Configuração 5](#_Toc495023321)

[Identificação da Configuração 5](#_Toc495023322)

[Métodos de Identificação 5](#_Toc495023323)

[Baselines do Projeto 5](#_Toc495023324)

[Tag 6](#_Toc495023325)

[Estrutura do Repositório de Versões 6](#_Toc495023326)

[Repositório principal 6](#_Toc495023327)

[Repositório de documentação (Wiki) 6](#_Toc495023328)

[Controle de Configuração e Mudança 8](#_Toc495023329)

[Processamento e Aprovação de Solicitações de Mudança 8](#_Toc495023330)

[Comitê de Controle de Mudança (CCB) 8](#_Toc495023331)

[Estimativa do Status de Configuração 8](#_Toc495023332)

[Processo de Armazenamento de Mídia e Liberação do Projeto 8](#_Toc495023333)

[Relatórios e Auditorias 9](#_Toc495023334)

[Marcos 11](#_Toc495023335)

[Treinamento e Recursos 11](#_Toc495023336)

# **Introdução e Finalidade**

O Plano de Gerência de Configuração do **QMS** apresenta todas as tarefas do Gerenciamento de Configuração e mudanças no projeto, para o controle das mudanças no decorrer das entregas das releases. Este documento Abrange toda a infra-estrutura utilizada nesse projeto.

### **Referências**

**i. SANTOS, Gustavo Moraes dos et al. Template de Plano de Gerenciamento de Configuração. Disponível em: http://github.com/gabrielaimeeg/DroidMetronome/wiki/TEMPLATE-Plano-de-Gerenciamento-de-Configuração. Acesso em: 19 Ago. 2017.**

# **Gerenciamento de Configuração de Software**

## **Organização, Responsabilidades e Interfaces**

Gerente de Configuração (GC):

* Carlos Henrique Matias Barbosa **@carlos-henrique28**

A gerência de configuração garante a consistência e a ligação de todas as atividades referentes ao projeto, prevenindo a inconsistência no desenvolvimento, assim o gerente de configuração é responsável pelo controle de todas as mudanças e disponibilizar corretamente a todos os envolvidos as versões e os itens de configuração do projeto (artefatos de software) para que sejam amenizadas falhas durante a evolução do desenvolvimento (auditoria das configurações).

#### **Artefatos da Gerência de Configuração**

As atividades da Gerência de Configuração estão definidas no MPS.BR nível F e são divididas em GCOs, que serão abrangidos e controlados nos artefatos gerados. Como o Scrum não contempla soluções nesta área explicitamente, adaptações foram feitas para que a gerência de configuração fosse implementada no processo.

| **Artefatos** |
| --- |
| Plano de Configuração |
| Repositório organizado para os Itens de Configuração |
| Banco de dados de Requisições de Mudança (Issues) |
| Termo de abertura do projeto |
| Documento de requisitos QMS |
| Plano do projeto QMS |
| Documento de arquitetura de software QMS |

### **Ferramentas, Ambiente e Infra-estrutura**

##### **Ferramentas**

| **Tipo** | **Ferramenta** | **Versão** |
| --- | --- | --- |
| Sistema Operacional | Elementary OS | 0.4.1 |
| Controle de Versão | GIT | 2.4.2 |
| Ambiente de Desenvolvimento (IDE) | Atom | v1.19.2 |
| Gerenciamento de Projetos | GitHub |  |
| Editor de Texto | GitHub Wiki |  |
| Editor de Diagramas | Bizagi Modeler |  |
| Controle de mudanças | GitHub Issues |  |
| Controle de versões | Google Drive |  |
| Comunicação | Skype / WhatsApp | 4.3 / 2.17.107 |

##### Requisitos mínimos de Hardware do Ambiente de Desenvolvimento

* Microprocessador celeron 1.4GHz
* Memória RAM 4GB
* Espaço livre em disco 30GB
* Conexão com a Internet.

##### Requisitos mínimos de Hardware do Ambiente de Execução (smartphone)

* Conexão com a Internet.

# **O Programa de Gerenciamento de Configuração**

## **Identificação da Configuração**

### **Métodos de Identificação**

Os artefatos criados ao decorrer do projeto, com exceção de código fonte, terão a seguinte nomenclatura como forma de padronização:

QMS\_<TEXTO\_LIVRE> - <TIPO\_ART>.<EXT>

| **Identificador** | **Descrição** |
| --- | --- |
| \*\*<TEXTO\_LIVRE>\*\* (Não obrigatório) | Texto livre que identifique unicamente o arquivo. Ex.: Reunião 01. |
| \*\*<TIPO\_ART>\*\* | Tipo do artefato. Ex: Diagrama de Caso de Uso, etc. |
| \*\*<EXT>\*\* | A extensão do arquivo. Ex.: doc, html, php, etc. |

### **Baselines do Projeto**

Para que seja iniciado o desenvolvimento de uma baseline, é necessário que o gerente de configuração faça uma Auditoria de Configuração primeiro e autorize a criação dessa baseline, haja vista que o mesmo estará envolvido em todas as etapas do projeto, para garantir o versionamento e gerenciamento de mudanças dessas baselines .

##### **Tag**

Como forma de padronização definida pela equipe, quando criadas, as baselines conterão todo o repositório Git, incluindo código fonte, diagramas, etc. e serão numeradas com o seguinte formato, a partir do número 0 (zero):

<major>.<minor>.<patch>.<feedback>

E serão construídas com as seguintes orientações:

* Indicador de início da primeira release, avança o <major>.
* Adição de novas funcionalidades sem quebrar compatibilidade com versões anteriores, avança o <minor>.
* Correção de bugs e outras alterações, avança <patch>.
* Quantidade de reprovações do cliente para cada release <feedback>

### **Estrutura do Repositório de Versões**

O GitHub é uma plataforma de hospedagem de projetos de software que oferece, além de armazenamento de repositórios ***GIT*** e/ou ***SVN***, uma área do repositório específica para documentação online (Portal Wiki) e outra para controlar tarefas e mudanças no projeto (issues). Por questões de versatilidade oferecidas pelo repositório, e por proporcionar as ferramentas que melhor auxiliam no controle do ciclo de vida RUP do **QMS** , todos os artefatos de software e o gerenciamento do projeto será feito no GitHub e/ou em ferramentas integradas ao GitHub.

##### **Repositório principal**

O repositório do **QMS**, podendo ser controlado pelo sistema de controle de versão [**GIT**](http://git-scm.com/) ou **SVN**, deve ter os seguintes artefatos nos respectivos diretórios relacionados abaixo (em ordem alfabética):

| **Diretório** | **Conteúdo** |
| --- | --- |
| **.** | • README do repositório |
| ***Artefatos*** | • Diretório possuindo todos os documentos referentes a todo o projeto desde o início |
| ***Planejamento*** | Diretório explicativo relacionado as atividades planejadas para desenvolvimento da release. |

##### Repositório de documentação (Wiki)

O Wiki do **QMS**, controlado apenas por **GIT** e pelo GitHub de forma online, **não tem** divisão por diretórios, pelo fato de que este repositório tem apenas arquivos de texto na linguagem de marcação Markdown (.md) convertidos automaticamente para páginas HTML pelo GitHub para exibição online no navegador de Internet. Como o GitHub cria uma abstração de que é um websitea documentação do repositório, não é exibido o formato de hierarquia de arquivos e pastas como no repositório principal.

Para estruturação do Wiki, a página principal (Home) e a barra lateral do Wiki terão a divisão feita por "**Portais**", mas para áreas de conhecimento da Engenharia de Software aplicadas no projeto:

| **Iniciação do Projeto** | **Documentação relacionada ao gerenciamento do projeto** |
| --- | --- |
| Termo de Abertura do Projeto | • Documento de iniciação do projeto |

| **Gerência de Projeto** | **Documentação relacionada ao gerenciamento do projeto** |
| --- | --- |
| Planejamento | • Plano de Projeto |
| Procedimentos | • Relatórios de Monitoramento |

| **Engenharia de Requisitos e Arquitetura** | **Documentação relacionada aos requisitos coletados para iniciação do projeto** |
| --- | --- |
| Planejamento | • Documentos referentes aos requisitos e arquitetura |

| **Gerência de Configuração** | **Documentação relacionada ao gerenciamento de itens de configuração do projeto** |
| --- | --- |
| Planejamento | • Plano de Gerência de Configuração e Mudanças |
| Procedimentos | • Relatórios de Auditorias de Configuração e Verificação dos itens de Mudança |

| **Garantia de Qualidade** | **Documentação relacionada ao controle de qualidade** |
| --- | --- |
| Planejamento | • Plano de Garantia da Qualidade |
| Procedimentos | • Relatórios de Verificação e Validação |

## **Controle de Configuração e Mudança**

### **Processamento e Aprovação de Solicitações de Mudança**

O repositório é mantido e controlado em conjunto com os Mantenedores do software. A plataforma de ***Issues*** do GitHub será utilizada para o controle das mudanças no software ao decorrer das releases.

Quando alguém cria uma nova issue (reporta um bug, solicita uma nova funcionalidade), é feita uma análise de viabilidade e riscos. Se a mudança solicitada não é viável, é informada ao autor da solicitação a inviabilidade e é fechada a issue; caso contrário, os Mantenedores planejam e executam a mudança e o Gerente de Configuração e o de Verificação e Validação monitoram o procedimento. Caso a mudança feita tenha sido um sucesso (revisada), o gerente de configuração prepara mais uma release (baseline) do software.

### **Comitê de Controle de Mudança (CCB)**

O Comitê de Controle de Mudança (CCB) será responsável por avaliar o impacto, benefícios, riscos associados, além de aprovar e priorizar as mudanças submetidas para a reunião do CCB. É formada pelo Gerente de Configuração, o Mantenedor de Software e o Gerente de Projeto e juntos farão a gestão de mudanças do projeto, para ter uma organização sobre as solicitações de mudanças no ***QMS***.

O gerente de projeto ficará responsável por redigir as Análises de Viabilidade e Riscos e o Planejamento de mudanças (caso sejam aprovadas), coordenar as reuniões, receber, organizar e comunicar informações referentes a mudanças e gerar a pauta, com antecedência de pelo menos um dia, e a ata com as decisões tomadas, disponibilizando-os em repositório único para manter um histórico de fácil acesso e consulta.

A solicitação de mudança (SM) deve ser deve ser registrada nas ***Issues*** do GitHub, assim como as Análises de Viabilidade e Riscos e o Planejamento de mudanças, disponível no repositório do projeto. Algumas informações são necessárias para a devida criação do solicitação.

## **Estimativa do Status de Configuração**

### **Processo de Armazenamento de Mídia e Liberação do Projeto**

O repositório do projeto deverá ser "clonado" por todos os integrantes da equipe de preferência em um diretório monitorado por um algum aplicativo de sincronização para nuvem (com Google Drive, e caso algum imprevisto aconteça em algum computador de um integrante da equipe ou com o repositório, o conteúdo poderá ser recuperado pelo serviço de armazenamento em nuvem no qual estava o diretório do repositório Git.

O processo de liberação do projeto diz respeito ao conteúdo do **release**, a quem ele se destina e se há quaisquer problemas conhecidos e instruções de instalação.

#### Relatórios e Auditorias

As auditorias, que tem como entrada-chave a saída das atividades de Verificação e Validação de software, serão realizadas **sempre antes** da liberação da baseline para o cliente, sendo responsável por verificar se o que está sendo liberado para o cliente está completo, no que tange as cláusulas contratuais, e correta, atendendo ao requisitos estabelecidos.

A auditoria será responsável por verificar se os componentes estão presentes nas versões especificadas e confirmar a presença de todos os artefatos necessários. Caso, durante a auditoria, alguma falha seja encontrada (Item de Configuração não encontrado), devem ser executados os seguintes passos de acordo com os padrões de gerenciamento do *RUP*:

| **Atividade** | **Descrição** | **Responsabilidade** |
| --- | --- | --- |
| Enviar CR | Qualquer envolvido no projeto pode enviar uma Solicitação de Mudança (CR). A solicitação é registrada no Sistema de Rastreamento de Solicitação de Mudança (por exemplo, o *issues* do wiki) e colocada na Fila de Revisão do CCB como Enviada. | Solicitante |
| Rever CR | A função desta atividade é rever as Solicitações de Mudança Enviadas. Uma revisão inicial do conteúdo da Solicitação de Mudança é feita na reunião de Revisão do CCB para determinar se a solicitação é válida. Se for, será decidido se a mudança está dentro ou fora do escopo dos releases atuais, de acordo com prioridade, programação, recursos, nível de esforço, risco, gravidade e outros critérios relevantes definidos pelo grupo. | CCB |
| Confirmar Duplicar ou Recusar | Se houver suspeita de uma Solicitação de Mudança estar Duplicada ou ter sido Recusada como inválida (por exemplo, erro do operador, não reproduzível, funcionamento etc), um representante do CCB terá que confirmar o estado da solicitação e obter mais informações do solicitante, se necessário. | Representante do CCB |
| Atualizar CR | Se houver necessidade de mais informações (Mais Informações) para a avaliação de uma Solicitação de Mudança ou se uma Solicitação de Mudança for recusada em algum momento durante o processo (por exemplo, foi confirmada como Duplicada, Recusada etc.), o solicitante será notificado e poderá atualizar a solicitação com as novas informações. A Solicitação de Mudança atualizada será reenviada para a Fila de Revisão do CCB para que os novos dados sejam avaliados. | Solicitante |
| Programar e Atribuir Tarefas | Depois que a Solicitação de Mudança é Aberta, o Gerente de Projeto atribui a tarefa ao membro da equipe adequado - de acordo com o tipo de solicitação (por exemplo, solicitação de melhoria, conserto de defeito, mudança de documentação, defeito de teste etc) - e faz as atualizações necessárias na programação do projeto. | Gerente de Projeto |
| Fazer Mudanças | O membro da equipe designado executará o conjunto de atividades definidas na seção adequada do processo (por exemplo, requisitos, análise e design, implementação, produzir materiais de suporte ao usuário, projetar testes) para efetuar as mudanças solicitadas. Essas atividades incluem todas as atividades normais de revisão e de testes unitários, como descrito no processo normal de desenvolvimento. A Solicitação de Mudança será, então, marcada como Resolvida. | Membro da Equipe Atribuído |
| Verificar Mudanças em Build de Teste | Depois de Resolvidas pelo membro da equipe atribuído (analista, desenvolvedor, testador, redator técnico), as mudanças são colocadas em uma fila de teste designada a um testador e Verificadas em um build de teste do produto. | Testador |
| Verificar Mudanças no Build do Release | Depois que a mudança resolvida é Verificada em um build de teste do produto, a Solicitação de Mudança é colocada em uma fila de releases que serão comparados a um build do release do produto, são produzidas notas de release etc, e a solicitação é Fechada. | Representante do CCB (Integrador do Sistema) |

# **Marcos**

Os principais marcos do projeto são os de último dia de sprint, em que há a criação da baseline do repositório de controle de versão e a realização das cerimônias de final de sprint do RUP. Mais informações no Plano de Projeto.

O Plano de Gerência de Configuração é alterado nos seguintes casos:

* O repositório de versões não está atendendo as necessidades dos integrantes da equipe;
* O Plano não foi aprovado pela ***V&V*** e precisa ser refatorado.

Quando for necessária a refatoração deste Plano, deve ser criada uma nova issue de **solicitação de mudança** no Plano de GCS.

# **Treinamento e Recursos**

* No README do repositório principal há um guia rápido para utilização do sistema de controle de versão Git;
* No site do GitHub há um tutorial iterativo para aprender a usar o Git em 15 minutos, criado pelo ***Code School***: https://try.github.io
* A documentação online do Git tem diversos conteúdos a respeito de controle de versão e gerência de configuração de forma completa, mas didática: http://git-scm.com/doc
* Nos guias online do GitHub tem um guia rápido para aprendizagem da linguagem de marcação Markdown: https://guides.github.com/features/mastering-markdown/